PAPER Special Issue on Knowledge-Based Software Engineering

Patterned Versus Conventional Object-Oriented Analysis Methods: A Group Project Experiment

Shuichiro YAMAMOTO† *and* **Hiroaki KUROKI**†, *Members*

SUMMARY Object-oriented analysis methods can be grouped into data-driven and behavior-driven approaches. With data-driven approaches, object models are developed based on a list of objects and their inter-relationships, which describe a static view of the real world. With behavior-oriented approaches, a system usage scenario is analyzed before developing the object models. Although qualitative comparisons of these two types of methods have been made, there was no statistical study has evaluated them based on controlled experiments. This paper proposes the patterned object-oriented method, POOM, which is a behavior-oriented approach, and compares it to OMT, a data-driven approach, using small team experiments. The effectiveness of POOM is shown in terms of productivity and homogeneity.
key words: *requirement analysis, object-oriented analysis, empirical study, role object model, object pattern*

1. Introduction

There are many object-oriented analysis methods [1]. Jacobson [2] categorizes them into two groups: data-driven approaches and behavior-driven approaches. With data-driven approaches, object models are developed based on a list of objects and their inter-relationships, which describe a static view of the real world. With behavior-oriented approaches, a system usage scenario is analyzed before developing the object models. Examples of data-driven approaches include OMT [3], Shlaer/Mellor [4], and Fusion [5]. Examples of behavior-oriented approaches include OBA [6], OOSE [7], and RDD [8].

If the real world can be sufficiently well defined, data-driven approaches provide robust object models. However, real-world objects are difficult to define for information systems because information does not directly correspond to concrete objects. With behavior-driven approaches, user roles and a usage scenario are initially extracted, the objects derived from the usage scenario. Because a scenario describes the external interactions and system boundaries, object models of behavior-driven approaches are likely to be stable. However, it is difficult to develop static data objects even though they are extremely important in information systems.

Sharble and Cohen [10] made a comparative study of two designs for a brewery problem. They compared data-driven and behavior-driven approaches quantitatively. They evaluated design level object models developed by Shlaer/Mellor and RDD methods for the problem of controlling the brewing process. They compared the numbers of attributes, methods, and messages for objects. The comparison showed that the Shlaer/Mellor approach (data-driven approach) is more complex than RDD (behavior-driven approach) because a centralized control object manages many data objects directly in the former approach, whereas autonomous objects collaborate with each other in the latter approach. Wirfs-Brock [11] examined their result and discussed how they differ.

This paper introduces a new behavior-driven approach, POOM, and compares it to OMT, a data-driven approach, for team analysis problems in the information system domain. Although Sharble and Cohen's discussion considered quantitative data, it was difficult to draw generic conclusions as the number of subjects was limited to only two. This paper describes statistical testing results based on experiments with 24 subjects. Table 1 summarizes the difference between Sharble and Cohen's approach and ours.

2. POOM

2.1 Object Model

Object patterns of POOM are shown Table 2. They were used to develop the patterned-object model, which describes the architectural objects from the view point of a three-tiered client-server architecture. In this architecture, the software components are separated into presentation, function, and data tiers. This hierarchical system structure makes client-server information systems

Table 1 Sharble and Cohen's to our approach.

Attribute	Sharble&Cohen's research	Our research
Methods compared	Shlaer/ Mellor vs. RDD	OMT vs. POOM
Object model	Design model	Analysis model
Number of subjects	2 subjects	24 subjects
Work style	Personal analysis	Team analysis
Application domain	Real time control domain	Information system domain

Manuscript received April 24, 1998.
Manuscript revised July 15, 1998.
†The authors are with NTT Multi-Media System Laboratory Group, Yokosuka-shi, 239–0847 Japan.

Table 2 Object patterns.

Category	Icon	Object type
Presentation tier		Role object type
		Owner object type
		Boundary object type
Function tier		Transaction object type
Data tier		Access object type
		Source object type
		Flow object type

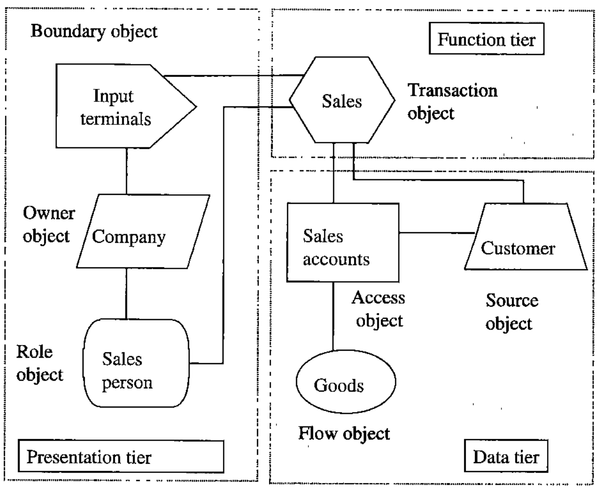

Fig. 1 Basic relationships of patterned objects.

more flexible because modifications are encapsulated in each tier. The patterned-object model defines the objects in the three tiers and their relationships. Seven basic object categories are used. The data tier includes the access object type, the flow object type, and the source object type. The function tier includes the transaction object type. The presentation tier includes the boundary object type, the owner object type, and the role object type.

The access objects are stored in the database and managed by the transaction objects. The flow objects represent the physical goods flowing in the real world. The source objects correspond to external events. For example, if a customer purchases a commodity, the customer is defined as a source object, the commodity as a flow object, and the record of purchasing the commodity as an access object. The boundary objects correspond to the user interfaces, such as display terminals. The role objects correspond to the roles of role-object models. The owner objects represent the owners of the boundary objects and of the role objects.

The basic relationships between the patterned objects are shown in Fig. 1.

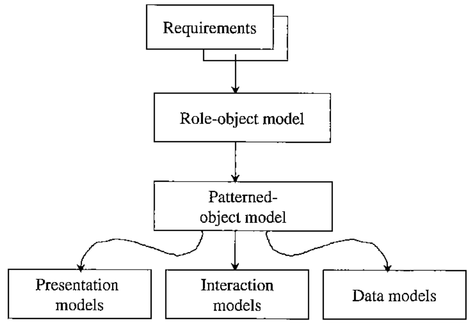

Fig. 2 Pattern-oriented object modeling method (POOM).

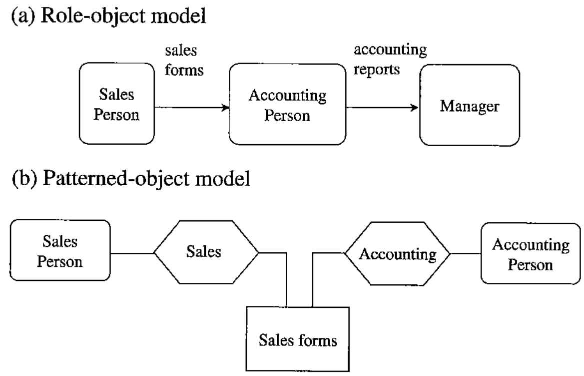

Fig. 3 Transformation from role-object model to patterned-object model.

2.2 Analysis Process

With the POOM method, the analysts initially analyze the role flows, which correspond to the scenarios of the behavior-driven approaches. They develop object models based on object patterns suitable for the architectures of client-server information systems. An overview of the POOM method is shown in Fig. 2. The POOM method uses two primary models: a role-object model and a patterned-object model.

(1) Role object modeling

The role-object model describes the roles of business persons and their relationships. The relationships are derived based on message sending from one role object to another role object. The relationships are defined by directed arcs connecting two role objects. An example of a role-object model is shown in Fig. 3 (a).

(2) Message analysis

The role-object model passes messages between role objects. Access objects can be derived from the data included in messages that specify the services of role objects. When the role object sends a message, a transaction object is necessary to create the access objects extracted from the message. Conversely, when the role object accepts a message, a transaction object is necessary to provide the services based on the access objects extracted from the message. In this way, transaction objects can be derived from access objects and role objects.

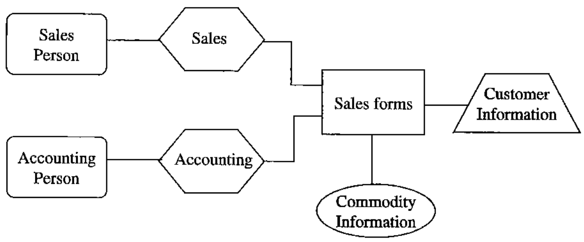

Fig. 4 Introduction of data tier objects.

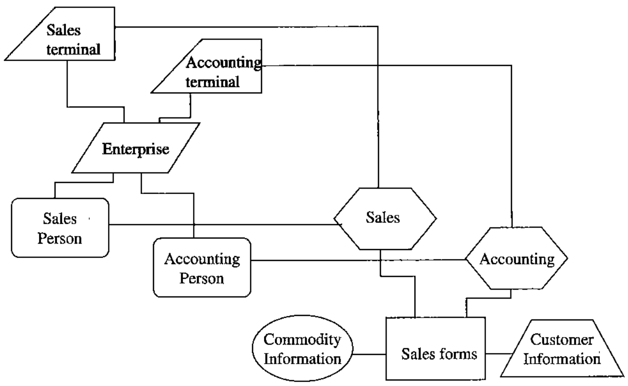

Fig. 5 Introduction of presentation tier objects.

Figure 3 (b) shows the patterned-object model introducing an access object and two transaction objects based on the role-object model of Fig. 3 (a).

(3) Data object analysis

By data object analysis, source objects and flow objects are added based on access objects. Figure 4 shows the object model developed from Fig. 3 (b). These objects can be extracted from the attributes of access objects.

(4) Presentation object analysis

Presentation object analysis allows owner objects and boundary objects to be introduced based on role objects and transaction objects. Figure 5 shows the object model introducing one owner object and two boundary objects based on Fig. 4.

(5) Refinement of object models

After developing patterned-object model, similarity among objects should be analyzed. The object model can be refined by extracting generic objects for each object type.

Figure 6 shows the object model refined from Fig. 5 with a generic boundary object and a generic transaction object.

3. Evaluation

To determine the effectiveness of POOM, we conducted a controlled experiment in which we evaluated POOM and OMT under the same conditions. An overview is shown in Fig. 7. In this paper, we compare the development of object models in OMT with that of those in POOM. The development of presentation models,

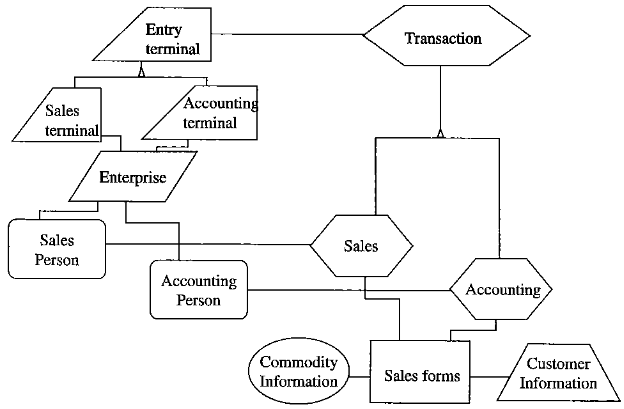

Fig. 6 Introduction of generic objects.

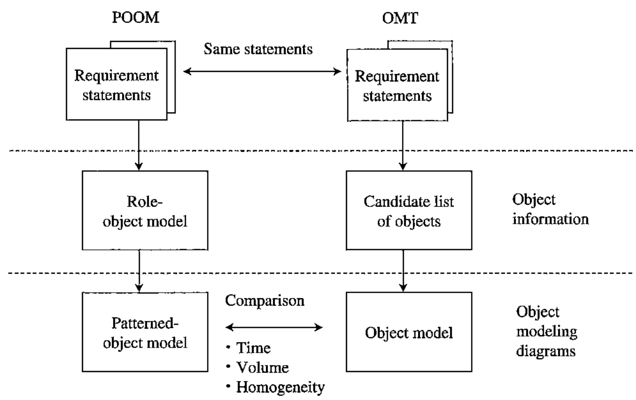

Fig. 7 Overview of experiments.

interaction models, and data models is not discussed.

3.1 Analysis Document

OMT and POOM were compared by extracting object information and object modeling diagrams as shown in Fig. 7. The products of OMT and POOM teams were almost same excepting the difference of extracting object information.

(1) Extracting object information

OMT teams developed object candidate lists based on a business scenario. POOM teams developed role-object models based on the same business scenario.

(2) Object modeling diagrams

OMT teams developed object models based on object candidate lists. POOM teams developed patterned-object models based on the role object models.

3.2 Schedule of Experiment

[Step1] Seminar of methods

The POOM group received three hours of lecture on the POOM method. The OMT group received three hours of lecture on the OMT method.

[Step2] Personal analysis experiment

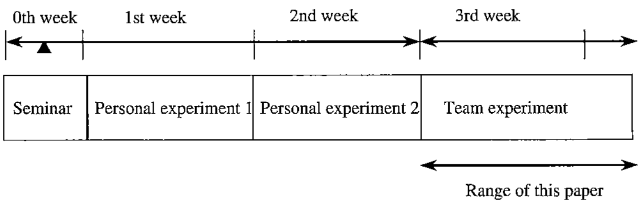

Fig. 8 Schedule of experiments.

Table 3 Team characteristics.

Team	OMT				POOM			
	A	B	C	D	A	B	C	D
No. of people	3	3	3	3	3	3	3	3
Experience in OMT	yes	no	no	no	yes	no	no	no
Avg. Programming experience (year)	4.33	7.33	6.67	5.67	6.0	5.33	10.3	4.67
Avg.	6.0				6.6			

In the personal analysis experiment, subjects were told to develop object models for the book store accounting problem (10 Japanese statements) and the travel agent problem (12 Japanese statements). Personal analysis experiments are not covered in this paper and their results will be described elsewhere.

[Step3] Team analysis experiment

Each of the teams analyzed the same problem: describe a sales management information system for selling liquor. The problem contains 87 statements in Japanese.

Figure 8 shows the schedule of the experiments.

3.3 Team Configuration

To evaluate the effectiveness of object oriented analysis methods, the 24 subjects were divided into two groups of POOM teams and OMT teams. The subjects of each team analyzed the same problem. Each team had three members. Table 3 shows the team configuration in the experiment. As 6 subjects had OMT experience, they were divided into 2 teams and were assigned to OMT and POOM groups. Average programming experience of each team is also shown in Table 3. T-tests showed that the average number of years experience in developing software was not significant.

3.4 Experimental Conditions

Because no tool supports the development of role-object and patterned-object models, word processors were used to develop the object models for both groups to make the experimental conditions the same. The OMT teams developed object lists and object-model diagrams. The POOM teams developed role-object models and patterned-object-model diagrams.

Table 4 Results of OMT teams.

Teams	OMT-A	OMT-B	OMT-C	OMT-D	Avg.
Object elicitation time	60	900	180	180	330
Object modeling time	900	1200	1500	240	960
No. of objects	39	11	31	40	30.25
No. of P-tier objects	5	0	5	5	3.75
No. of F-tier objects	17	6	7	14	11
No. of D-tier objects	17	5	19	21	15.5
No. of majority objects	7	1	7	6	5.25
No. of P-tier majority objects	3	0	3	3	2.25
No. of F-tier majority objects	0	0	0	0	0
No. of D-tier majority objects	4	1	4	3	3
No. of relationships	62	23	63	45	48.25
Ratio of majority objects	0.1795	0.0909	0.2258	0.15	0.1616
Productivity	6.3125	0.9714	3.3571	12.75	3.6654

Table 5 Results of POOM teams.

Teams	POOM-A	POOM-B	POOM-C	POOM-D	Avg.
Object elicitation time	270	75	180	300	206.25
Object modeling time	300	480	300	600	420
No. of objects	68	43	45	54	52.5
No. of P-tier objects	12	8	11	10	10.25
No. of F-tier objects	20	15	12	16	15.75
No. of D-tier objects	36	20	22	28	26.5
No. of majority objects	36	34	35	38	35.75
No. of P-tier majority objects	11	7	11	10	9.75
No. of F-tier majority objects	12	15	12	15	13.5
No. of D-tier majority objects	13	12	12	13	12.5
No. of relationships	103	67	71	74	78.75
Ratio of majority objects	0.5294	0.7907	0.7778	0.7037	0.7004
Productivity	18.0	11.892	14.5	8.5353	12.575

4. Experimental Results

4.1 Analysis Data

The experimental results for OMT and POOM are summarized in Tables 4 and 5, respectively. These tables show the analysis times, numbers of objects, and relations of object models. Also, the number of objects in each of the three tiers is listed. The number of common objects for a team is defined as the number of objects commonly extracted by most of the teams. Here, two objects in different object models are viewed as equivalent if they have the same name. The notion of majority object was created by the authors to evaluate the similarity of object models among different teams. The number of majority objects is the number of objects commonly extracted by more than the half of analysis teams in the same method group.

4.2 Comparison of the Analysis Time

[Observation 1] The mean time for developing POOM object-model diagrams was smaller than that of the OMT ones (significant at the 0.05 level).

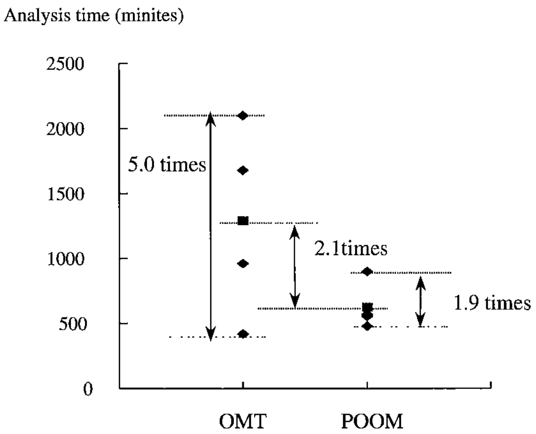

Fig. 9 Comparison of analysis time.

The mean times for developing the POOM and OMT object models were 10.4 and 20.3 hours, respectively. The POOM analysis time was about 51.2% that of OMT time. The mean difference is significant at the 0.05 level.

The analysis times of the OMT and POOM teams are shown in Fig. 9.

4.3 Comparison of the Number of Objects and Relations

[Observation 2] The mean number of objects in the POOM object-model diagrams was greater than those in the OMT ones (significant at the 0.05 level).
The mean number of POOM objects was approximately 52.5, although the mean number of OMT objects was approximately 30.3. The number of OMT objects was about 57.7% that of the POOM ones. The mean difference is significant at the 0.05 level.
[Observation 3] The mean number of relations in the POOM object-model diagrams was greater than that of the OMT ones (significant at the 0.05 level).
The mean numbers of relations were 78.8 and 48.3, respectively. The number of OMT relations was about 61.3% that of the POOM ones. The mean difference is significant at the 0.05 level.

4.4 Comparison of Productivity

[Observation 4] The productivity of the POOM teams was higher than that of the OMT ones.
Productivity is defined as the number of objects and relations developed per hour. The productivity of POOM and OMT was 13.2 and 5.8, respectively. The productivity of the POOM teams was approximately twice that of the OMT ones. The mean difference is statistically significant.
[Observation 5] The variance in productivity of the POOM teams was smaller than that of the OMT ones (significant at the 0.05 level).
The productivity of the OMT and POOM teams are shown in Fig. 10. The maximum difference of the productivity of OMT teams is about 13 times, whereas that of POOM ones is only 2 times.

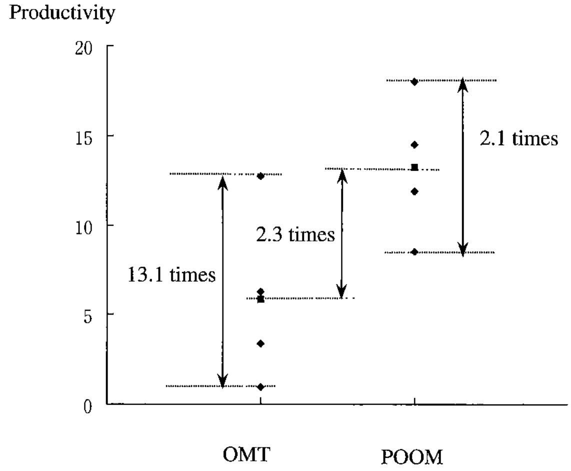

Fig. 10 Comparison of productivity.

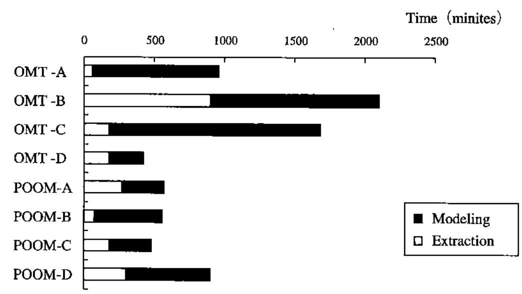

Fig. 11 Comparison of time required for analysis tasks.

4.5 Comparison of Analysis Tasks

Figure 11 shows the time taken by each team for the object extraction and object modeling tasks. The average ratio of object extraction and object modeling time for the OMT teams was 1 to 3. That of the POOM ones was 1 to 2.
[Observation 6] The ratio of the number of objects in object models to the number of extracted objects of the POOM teams was greater than that of the OMT ones (significant at the 0.05 level).

Figure 12 shows the relationship between the number of extracted and modeled objects. The number of extracted objects of the POOM teams was one twentieth that of the OMT ones. The number of developed objects of the OMT teams was two thirds that of the POOM ones.

4.6 Comparison of Homogeneity

[Observation 7] The number of POOM majority objects was greater than that of the OMT ones (significant at the 0.05 level).

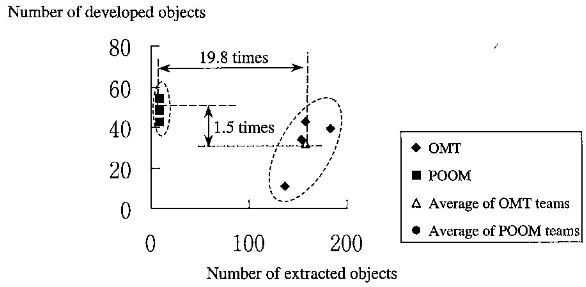

Fig. 12 Relationship between extracted objects vs. developed objects.

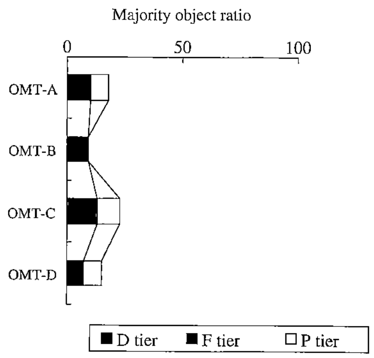

Fig. 13 Majority object ratio of OMT teams.

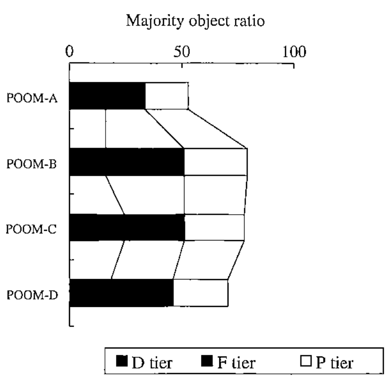

Fig. 14 Majority object ratio of POOM teams.

[Observation 8] The number of POOM majority objects in each tier was greater than that of the OMT ones (significant at the 0.05 level).
The numbers of the OMT and POOM majority objects are shown in Figs. 12 and 13, respectively.

The mean numbers of majority objects were 35.8 and 5.3, respectively. Therefore, the mean number of majority objects of the POOM teams was about 6.7 times greater than that of the OMT teams.

The mean difference in the number of majority objects between the POOM and OMT teams is significant at the 0.05 level. This also holds for the number of majority objects in each tier.

Figures 13 and 14 show the majority object ratio of the OMT teams and the POOM teams. The majority

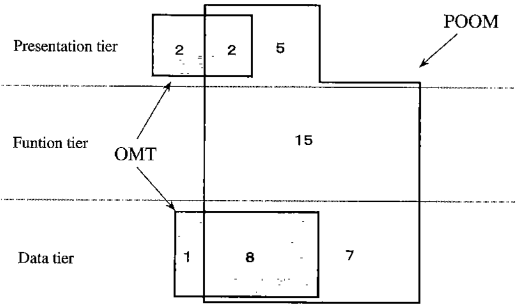

Fig. 15 Comparison of number of majority objects.

object ratio for each tier of the OMT teams was small on the whole; in particular there was no majority object in the function tier. The ratio for each tier of the POOM ones was large on the whole.
[Observation 9] 77% of the majority objects of OMT teams were also discovered by the POOM ones.

Figure 15 shows the inclusion relationship of majority objects in each tier. 10 majority objects were commonly developed by the OMT and POOM teams. The majority objects developed by only the OMT teams were the traders and the administrators in the presentation tier, and the sales person management information in the data tier. 27 majority objects of the POOM teams were not found by the OMT ones. The traders information was included as the data tier majority object of the POOM teams. As the administrators were operators of the external administration system, the administrators should not be included in the object models for the target system. The sales person information was included in the attributes of the sales information object of the POOM teams. Therefore, the essential majority objects of the OMT teams were all included in those of the POOM ones.

4.7 Evaluation by Questionnaire

[Observation 10] The opinion evaluation point score of POOM teams was greater than that of OMT ones (significant at the 0.05 level).
A series of questions were used to compare OMT and POOM after the experiment. The following questionnaire was answered yes or no by subjects.
[Q1] Was your creativity stimulated by the method?
[Q2] Did you analyze object models straightforwardly by the method?
[Q3] Does the method make the object model highly traceable to the architectural design?
[Q4] Did you analyze object models objectively by the method?
[Q5] Were your object models understandable by the method?
[Q6] Is the object model highly traceable to the business scenario by the method?

Figure 16 shows the results of the questionnaire

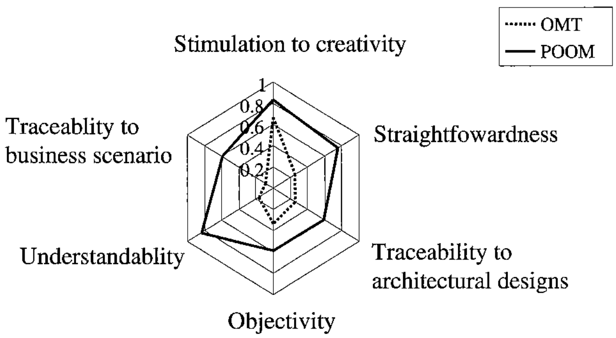

Fig. 16 Evaluation by questionaries.

for the 24 subjects. More than half of the POOM subjects responded positively to all questions. On the other hand, more than the half of the OMT subjects responded negatively to the questions except for Q1. The average of number of positive answers of POOM and OMT subjects were 4.17, and 1.75, respectively.

5. Considerations

5.1 Effectiveness

Table 6 summarizes the results of the statistical comparison of POOM and OMT. This shows the effectiveness of POOM compared to OMT. The relationship between productivity and the majority object ratio is shown in Fig. 17. This shows that POOM is superior to OMT in the terms of productivity as well as homogeneity. This is because POOM teams derive the objects from the role-flows of a usage scenario. Also, it is easy to develop object models as the categories of the POOM objects are clearly defined.

5.2 Limitations

(1) Figure 18 shows the majority object ratio of the POOM and the OMT teams for each tier. The majority object ratio of the POOM teams was not so high in the data tier. This may because data tier objects can not be completely extracted from the relationships between the role objects. When applying POOM to complex problems, there is a possibility that the homogeneity of the data tier may decrease. To solve this problem, it is necessary to develop a systematic analysis process for data tier objects.
(2) In this paper, the patterned-object model used seven object types created for the information system domain. For an other application domain, there may be some difficulty in modeling objects into the seven object types. In such a case, however, it is possible to extend the patterned-object model by creating new kinds of object types.
(3) In the experiment, contributions of the POOM object types and the POOM analysis process based on the object types were not measured separately. To clarify each contributions, it is necessary to compare the development of POOM object models with and without POOM analysis process described in Sect. 2.2.
(4) As the subjects of the experiment were not have object oriented programming experience, the result of the paper was open to the experienced object oriented programmers. Future research is necessary to evaluate the effectiveness of POOM for OOP experts. However, this paper provides the useful result to apply object oriented analysis method for engineers who have traditional programming skills.
(5) In this paper, the number of majority object is based on the name equivalence of objects. There are other kinds of object equivalence as follows.
[Attribute equivalence] Two objects in different object models are attribute equivalent if they have the same

Table 6 Results of statistical testing of POOM vs. OMT.

Criteria	Results	levels of significance
Avg. analysis time	OMT > POOM	5%
Avg. number of objects	OMT < POOM	5%
Avg. number of relations	OMT < POOM	5%
Avg. majority object ratio	OMT < POOM	5%
Avg. productivity	OMT < POOM	10%
Avg. point of opinion evaluation	OMT < POOM	5%

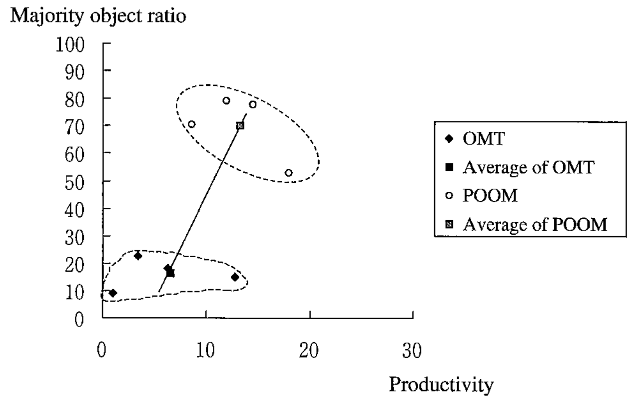

Fig. 17 Relationship between productivity and majority object ratio.

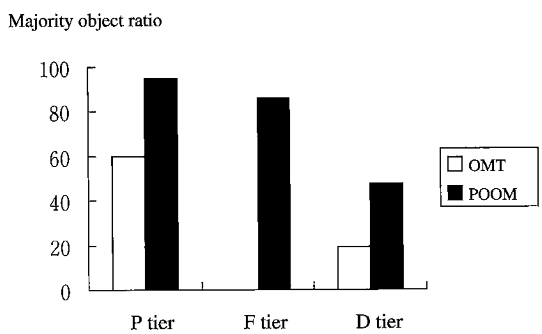

Fig. 18 Comparison of majority object ratio.

name and the same set of attributes.
[Method equivalence] Two objects in different object models are method equivalent if they have the same name and the same set of methods.
[Relationship equivalence] Two objects in different object models are relationship equivalent if they have the same name and the same set of relationships.

It is possible to extend the notion of majority object. Future research is needed to evaluate the effectiveness of POOM based on these kinds of object equivalence.

5.3 Stimulation of Creativity

More than the half of subjects of OMT teams answered positively for the Q1 of the questionnaire, although other questions were answered negatively by them. The reason of this is that OMT teams freely extracted objects using object list in the first step of the analysis. As subjects had different views of points to extract objects, it was not easy to understand object models developed by other team members. This may be the reason why other questions were answered negatively by OMT subjects.

6. Conclusion

Using a controlled experiment to compare our pattern-oriented object-modeling method, POOM, with OMT, we found that POOM is superior in terms of productivity and homogeneity. We have also conducted experiments on reusing object models. The results will be reported in elsewhere.

Acknowledgments

Authors are grateful to Mr. Masao Itoh for providing the references on the stereotyping approaches [9], [11]. Authors also thank reviewers for their helpful comments to refine the paper.

References

[1] D. Monarchi and G. Puhr, "A research typology for object-oriented analysis and design," CACM, vol.35, no.9, pp.35–47, 1992.
[2] I. Jacobson, "A confused world of OOA and OOD," JOOP, vol.8, no.9, pp.15–20, 1995.
[3] J. Rumbaugh, et al., "Object-Oriented Modeling and Design," Prentice Hall, Englewood Cliffs, NJ, 1991.
[4] S. Shlaer and S. Mellor, "Object-Oriented Systems Analysis: Modeling the World in Data," Prentice Hall, 1988.
[5] D. Coleman, et al., "Object-Oriented Development: The Fusion Method," Prentice Hall, 1994.
[6] K. Rubin and A. Goldberg, "Object behavior analysis," CACM, vol.35, no.9, pp.48–62, 1992.
[7] I. Jacobson, "Object-Oriented Software Engineering, A Use Case Driven Approach," ACM Press, Addison-Wesley, 1992.
[8] R. Wirfs-Brock and L. Wiener, "Designing Object-Oriented Software," Prentice Hall, 1991.
[9] R. Wirfs-Brock, "Stereotyping: A technique for characterizing objects and their interactions," Object Magazine, vol.3, no.4, pp.50–53, 1993.
[10] R.C. Sharble and S.S. Cohen, "The object-oriented brewery: A comparison of two object-oriented development methods," ACM SIGSOFT Software Engineering Notes, vol.18, no.2, pp.60–73, 1993.
[11] R. Wirfs-Brock, "How designs differ," ROAD, vol.1, no.4, pp.51–56, 1994.

Shuichiro Yamamoto is a senior research engineer, supervisor, of NTT Multi-Media System Laboratory Group. Previously, he was engaged in the development of CASE tools and distributed application development platforms. His research interests include distributed information systems, end user computing, reengineering, and requirements engineering. Mr. Yamamoto received a B.S. in information engineering from Nagoya Institute of Technology in 1977, and an M.S. in information engineering from Nagoya University in 1979. He is a member of IEEE, ACM, IPSJ and JSAI.

Hiroaki Kuroki is a manager of NTT Multimedia Business Department. Previously, he was engaged the development of the software development environments. His research interests include interactive multimedia contents on Web. Mr. Kuroki received a B.S. in electrical engineering and a M.S. in engineering from University of Osaka Prefecture in 1982. He is a member of IPSJ.